Homework 2

CMSY-217, Spring 2014

Upload your solution to the Canvas course website as a zip archive file prior to the start of class on Thursday, March 6.

1. Write a Java class called `Canasta` which belongs to the package which is the reverse of your HCC email address. For example, my `Canasta` class would belong to the package `edu.howardcc.mikemiller`.

2. Make the following modifications to the classes of Fig. 20.10:

   (a) Move the `Card` class to a separate file and change its access modifier to `public`

   (b) Move the `main` method from the `DeckOfCards` class to the `Canasta` class

3. Make the following modifications to the `Card` class:

   (a) Implement a `Comparable<Card>` interface where the natural ordering is based on the face of each card and corresponds to the order in which they appear in the `Face` enumeration

   (b) Add the capability to create `Card` objects which represent Jokers and make them the largest in the natural ordering

   (c) For the `toString` method, a Joker should simply return the string `Joker`

   (d) Add a method to return an `int` which represents the point value of each card based on the following table

Card	**Point Value**
red 3 (hearts or diamonds)	100
Joker	50
A, 2	20
8, 9, 10, J, Q, K	10
black 3 (clubs or spades), 4, 5, 6, 7	5

4. Make the following modifications to the `DeckOfCards` class:

   (a) Add a one-argument constructor which takes an `int` indicating the number of Jokers in the deck

   (b) Write a no-argument constructor which calls the one-argument constructor to create a deck with no Jokers

   (c) Instead of obtaining a `List` view of the array in the constuctor, pass `Arrays.asList(deck)` to the `LinkedList` constructor to create a new `List`

(d) Write a method called `combine` which takes a `DeckOfCards` object as an argument, combines the current deck with the argument, shuffles the combined deck, and then returns the combined deck

(e) Write a method which deals one card from the top of the deck; It should actually remove the `Card` from the instance variable `list` and return it to the caller

5. Write class called `Hand` which represents a hand that been dealt to a player. It's internal design might be similar to the `DeckOfCards` class. The `Hand` class should include methods to add a card, sort the cards, return the total of the point values of the cards, and print the cards in the hand.

6. Write code in the `main` method of the `Canasta` class for a four-player game of Canasta. You are *not* required to implement the entire game. At a minimum, you should at least do the following:

   (a) Create two decks of cards with two Jokers in each

   (b) Combine the two decks into one and shuffle the combined deck

   (c) Create an empty hand for each of the four players

   (d) Deal the cards, one card at a time to each player, until each of the four players has eleven cards

   (e) Print the cards each of the four hands along the the total point value of the cards in each hand

   Make use of objected-oriented programming techniques in your design and utilize the classes and methods from the Java Collections Framework as much as possible.

7. The rules for Canasta are available online. Two excellent sources of information are: http://games.yahoo.com/help/rules/ca; and http://en.wikipedia.org/wiki/Canasta#Rules_for_original_Canasta

Sample output which demonstrates the minimum requirements:

```
Player 1: 245
Ace of Hearts      Deuce of Clubs     Three of Hearts    Four of Clubs
Four of Spades     Four of Clubs      Eight of Clubs     Ten of Clubs
King of Spades     King of Diamonds   Joker

Player 2: 105
Ace of Hearts      Deuce of Clubs     Four of Hearts     Five of Clubs
Five of Hearts     Seven of Clubs     Seven of Clubs     Ten of Diamonds
Queen of Clubs     Queen of Diamonds  Queen of Spades

Player 3: 125
Deuce of Diamonds  Deuce of Hearts    Deuce of Spades    Five of Hearts
Five of Diamonds   Six of Spades      Nine of Clubs      Nine of Diamonds
Jack of Hearts     King of Diamonds   King of Hearts

Player 4: 195
Ace of Spades      Deuce of Hearts    Three of Clubs     Four of Hearts
Seven of Spades    Eight of Spades    Nine of Hearts     Ten of Hearts
King of Hearts     Joker              Joker
```